Improved approximation bounds for Vector Bin Packing

Chetan S. Rao
Department of Computer Science & Engineering
National Institute of Technology Calicut
Kerala 673601, India

Jeffrey John Geevarghese
Department of Computer Science & Engineering
National Institute of Technology Calicut
Kerala 673601, India

Karthik Rajan
Department of Computer Science & Engineering
National Institute of Technology Calicut
Kerala 673601, India

July 9, 2010

Abstract

In this paper we propose an improved approximation scheme for the Vector Bin Packing problem (VBP), based on the combination of (near-)optimal solution of the Linear Programming (LP) relaxation and a greedy (modified first-fit) heuristic. The Vector Bin Packing problem of higher dimension ($d \geq 2$) is not known to have asymptotic polynomial-time approximation schemes (unless P = NP).

Our algorithm improves over the previously-known guarantee of (ln $d + 1 + \epsilon$) by Bansal et al. [1] for higher dimensions ($d > 2$). We provide a $\theta(1)$ approximation scheme for certain set of inputs for any dimension d. More precisely, we provide a 2-OPT algorithm, a result which is irrespective of the number of dimensions d.

1 Introduction

Packing items of variable sizes into a given space is a fundamental problem of combinatorial optimization. This problem dates back to the origin of Operations Research. The packing problem and it's multidimensional variants have vital practical applications in diverse domains including the cutting, packaging and other industries. In this paper, we consider a type of packing problem called the Vector Bin Packing (Vector Packing) problem and propose a better bound for the multidimensional version of the problem. The items of variable sizes are packed into containers called bins with fixed size along all dimensions.

Multidimensionality plays an important role in capturing incomparable characteristics of the objects that are to be packed. For example, the memory requirements and bandwidth requirements in a distributed computing environment are incomparable. Multidimensionality also suffices the different costs that may be associated with each of these requirements. In this paper, we provide a method which is applicable to any dimension d.

1.1 Problem Definition

We now formulate the optimization problem that we are addressing.

Vector Bin Packing problem (VBP)
Given a set S of n d-dimensional vectors p_1, p_2, ..., p_n from [0,1]^d, find a packing (partition) of S into A_1, A_2, ..., A_m such that $\sum_{p \in A_i} p^k \leq 1$, $\forall i, k$ (p^k denotes the projection of vector p along 'k'th dimension). The objective is minimize the value of m, the number of partitions.
When $d = 1$, the problem is an instance of the classical bin packing problem (BP).

1.2 Related work

One dimensional bin packing problem has been studied extensively. Fernandez de la Vega and Lueker [7] gave the first asymptotic polynomial-time approximation scheme (APTAS). They put forward a rounding technique that allowed them to reduce the problem of packing large items to finding an optimum packing of just a constant number of items (at a cost of ϵ times OPT). Their algorithm was later improved by Karmarkar and Karp [9], to a ($1+log^2$)-OPT bound.

For 2-dimensional vector bin packing, Woeginger [12] proved that there is no APTAS. For higher dimensions, Fernandez de la Vega and Lueker [7] proposed a simple $(d + \epsilon)$-OPT algorithm, which extends the idea of 1-dimensional bin packing. Chekuri and Khanna [5] showed an $O(\log\ d)$-approximation algorithm that runs in polynomial time for fixed d. Bansal et al. [1] improved this result, showing an (ln $d + 1 + \epsilon$)-approximation algorithm for any $\epsilon \geq 0$. Karger et al. [8] have recently proposed a polynomial approximation scheme for randomized instances of the multidimensional vector bin packing using smoothing techniques. Patt-Shamir et al. [11] have recently explored the vector bin packing problem with bins of varying sizes and propose a (ln $2d + 1 + \epsilon$)-approximation algorithm for any $\epsilon \geq 0$.

1.3 Our results and organization

In this paper, we provide an improved approximation bound for the vector bin packing problem of any dimension. We provide a $\theta(1)$ approximation guarantee for certain set of inputs for any dimension d. In more specific settings, we provide a 2-OPT guarantee for large inputs irrespective of the dimension d. This is a notable improvement over the previously known guarantee of $(ln\ d + 1 + \epsilon)$ for higher dimensions $(d > 2)$.

The rest of the paper is organized as follows. Section 2 gives the linear program (LP) formulation of the problem. In Section 3, we provide the approximation algorithm and prove the bounds.

2 Problem formulation

We formulate the problem as an integer program in 2.1. The integer constraints are relaxed and we formulate it's dual (2.2). The solution of the relaxed integer program gives a thoughtful insight about the optimal number of bins.

2.1 Integer Linear Program (ILP) formulation

The vector bin packing problem (VBP) can be formulated as an integer program. We use two binary variables x_{ij} and y_j. The binary variable x_{ij} indicates if vector p_i is assigned to bin j and the binary variable y_j indicates whether bin j is in use or not. Our objective is to minimize the number of bins used.

The number of bins m can initially be set to a sufficiently large value arrived at by any heuristic (example - de la Vega and Leuker [7] give a $O(d)$-OPT bound on the number of bins). Then, we formulate the integer program (ILP) as follows -

$$
\begin{aligned}
&\text{minimize}: && \sum_j y_j && && (1)\\
&\text{such that} && \sum_j x_{ij} = 1 && 1 \le i \le n && (2)\\
& && \sum_i p_i^k . x_{ij} \le 1 && 1 \le j \le m, 1 \le k \le d && (3)\\
& && y_j \ge x_{ij} && 1 \le i \le n, 1 \le j \le m && (4)\\
& && x_{ij} \in \{0,1\} && 1 \le i \le n, 1 \le j \le m && (5)
\end{aligned}
$$

The constraints of the ILP are as follows -

- Constraint (2) states that every vector is packed in a bin.
- Constraint (3) ensures that the packed vectors do not exceed the bin dimensions.
- Constraint (4) tells whether a bin is used or not.
- Constraint (5) ensures that a vector is either packed entirely in a bin or not.

Constraint (5) can be relaxed as follows to obtain a linear program (LP).

$$x_{ij} \ge 0 \qquad 1 \le i \le n, \quad 1 \le j \le m \tag{5a}$$

We can obtain a feasible solution for the LP using any standard method [6]. Using binary search technique, we can also find the least value of m, $m^{'} \in Z^{+}$ for the relaxed ILP for which a feasible solution exists. The value of $m^{'}$ thus obtained will be less than the optimal solution for the integer program i.e. ($m^{'} \le$ OPT). However, the solution obtained is usually not integral. To tackle this problem, we formulate a dual-maximization problem for the above relaxed ILP.

Theorem 1. *[5] Any basic feasible solution to the relaxed LP defined by Equations 2, 3, and 5a has at most $d \cdot m$ vectors that are fractionally assigned to more than one bin.*

2.2 Dual-maximization problem

We introduce several new variables z_{ij} to formulate the dual. The dual-maximization problem formulation is given in the Appendix 4.1. We arrive at the following set of equations and constraints -

$$\text{maximize : } \sum_i \sum_j x_{ij} z_{ij} \tag{6}$$

$$\text{such that} \quad \sum_j x_{ij} = 1 \qquad 1 \le i \le n \tag{7}$$

$$\sum_i p_i^k . x_{ij} \le 1 \qquad 1 \le j \le m, 1 \le k \le d \tag{8}$$

$$\sum_i z_{ij} \le 1 \qquad 1 \le j \le m \tag{9}$$

$$x_{ij}, z_{ij} \ge 0 \qquad 1 \le i \le n, 1 \le j \le m \tag{10}$$

This is a nonlinear program (NLP) as the objective function is nonlinear. Hereafter, we shall refer to it as NLP.

Lemma 2. *The value of the objective function of the nonlinear program (NLP) lies between 1 and m.*

Proof.
Lower bound : Let $z_{ij} = \frac{1}{n} \forall (i,j)$. Then, the objective function becomes -

$$\text{Obj} = \sum_i \sum_j x_{ij} \frac{1}{n} = \frac{1}{n} (\sum_i \sum_j x_{ij}) = \frac{1}{n}(n) = 1$$

Upper bound : The value of the objective function is upper bounded as follows -

$$\text{Obj} = \sum_i \sum_j x_{ij} z_{ij} \le \sum_i \sum_j z_{ij} = \sum_j (\sum_i z_{ij}) \le \sum_j (1) = m$$

Thus, the range of values the objective function will assume is between 1 and m. ∎

The actual range of values that $\sum_i \sum_j x_{ij} z_{ij}$ (given the constraints) is 0 to m. The objective function gives us a hint about the utility of the bins. Intuitively, if the value of the objective function is k, we can assume that the expected utility of each bin is $\frac{k}{m}$.

Lemma 3. *In the optimal solution to the nonlinear program (NLP), if any values of x_{ij}'s are 0, the corresponding values of z_{ij}'s are 0.*

PROOF. Let $x_{i_1 j_1} = 0$ for some value of (i_1, j_1). From constraint (9) of the nonlinear program, we have -

$$\sum_i z_{ij_1} = z_{1j_1} + z_{2j_1} + \ldots + z_{i_1 j_1} + \ldots + z_{nj_1} \leq 1$$

Atleast one vector $i (i \neq i_1)$ is packed in bin j_1 (i.e. $x_{ij_1} > 0$). For this vector i, the value of $z_{ij_1} > 0$ in the optimal solution. The term $\sum_i x_{ij_1} z_{ij_1}$ is maximized iff the value of $z_{i_1 j_1}$ is redistributed among other z_{ij_1} values where $x_{ij_1} > 0$. Hence, the value of $z_{i_1 j_1}$ must be 0. ∎

Theorem 4. *Given a sufficiently large value of $m (m \geq OPT)$, the solution (value of x_{ij}) returned by the nonlinear program NLP will be integral, given that z_{ij} is non-zero whenever $x_{ij} \neq 0$.*

PROOF. Consider n vectors which are to be packed. Assume that the the optimal packing can be done with 2 bins (i.e. OPT=2). Without loss of generality, let us assume that the vectors $1, 2, \ldots, k_1$ ($k_1 < n$) be packed in bin 1, and vectors $k_1 + 1, k_1 + 2, \ldots, n$ be packed in bin 2 in the optimal solution OPT.
That is, let $x_{11} = x_{21} = \ldots = x_{k_1 1} = x_{(k_1+1)2} = \ldots = x_{n2} = 1$. The maximum value of the objective function in this case will be -

$$\begin{aligned} \text{Obj} &= \sum_{i=1}^{k_1} z_{i1} + \sum_{i=k_1+1}^{n} z_{i2} \\ &\leq 1 + 1 = 2 \end{aligned}$$

Now, let us assume that the vector $k_1 + 1$ be fractionally packable in bin 1 and bin 2. Let the fraction that can be packed be $t (0 < t < 1)$ in bin 1 and $(1 - t)$ in bin 2. The objective function in this case can be shown to be -

$$\begin{aligned} \text{Obj} &= \sum_{i=1}^{k_1} z_{i1} + t z_{(k_1+1)1} + \sum_{i=k_1+2}^{n} z_{i2} + (1-t) z_{(k_1+1)2} \\ &\leq 1 - (1-t) z_{(k_1+1)1} + 1 - t z_{(k_1+1)2} \\ &\leq 2 - (1-t) z_{(k_1+1)1} - t z_{(k_1+1)2} \\ &< 2. \end{aligned}$$

Similarly, we can extend the same argument when more than 1 vectors from bin 2 are fractionally packed in bin 1. The objective function strictly decreases as follows -

$$\begin{aligned}\text{Obj} &= \sum_{i=1}^{k_1} z_{i1} + \sum_{i=k_1}^{n} t_i z_{i1} + \sum_{i=k_1+1}^{n} (1-t_i) z_{i2} \\ &\leq 1 - \sum_{i=k_1+1}^{n} (1-t_i) z_{i1} + 1 - \sum_{i=k_1+1}^{n} t_i z_{i2} \\ &\leq 2 - \sum_{i=k_1+1}^{n} (1-t_i) z_{i1} - \sum_{i=k_1+1}^{n} t_i z_{i2} \\ &< 2.\end{aligned}$$

where t_i and (1-t_i) are the fractional packings of vector i in bin 1 & 2 respectively.

By induction, the same proof can be extended when the number of required bins is more than 2 (i.e. OPT > 2). Thus, the maximum value of the objective function occurs when the variables x_{ij} are integral. ∎

Theorem 5. *For the solution to be (near-) optimal and integral, the value of z_{ij} should be a function of x_{ij}.*

PROOF. The integral part follows from Theorem 4. Let us look into the optimality of the solution obtained.
From the Jensen's Inequality, we have that if f is a convex function ("concave-up") on an interval I and $a_i \in I$ then for weights λ_i summing to 1 -

$$f(\sum_{i=1}^{n} \lambda_i a_i) \leq \sum_{i=1}^{n} \lambda_i f(a_i)$$

We can apply Jensen's inequality with λ_i and a_i corresponding to x_{ij} and z_{ij}, respectively. The modified set of equations in this case is as follows -

$$f(\sum_{j=1}^{m} x_{ij} z_{ij}) \leq \sum_{j=1}^{m} x_{ij} f(z_{ij}) \qquad 1 \leq i \leq n \tag{a}$$

From the property of convex functions, we have -

$$f(tx) \leq t f(x) \qquad 0 \leq t \leq 1 \tag{b}$$

From (a) and (b), we have -

$$f(\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}z_{ij}) \leq \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} x_{ij}f(z_{ij}) \tag{c}$$

Since $f(x)$ is a convex function, any value which maximizes x also maximizes $f(x)$ and vice-versa. Hence, from the inequality (c), we have that the term $x_{ij}f(z_{ij})$ should be maximized for the objective function to be maximized. Indirectly, z_{ij} has to be maximized relative to the values of x_{ij}. The value of z_{ij} is constrained by the constraint (9), and hence we come up with the following function for z_{ij} -

$$z_{ij} = \frac{x_{ij}}{\sum_i x_{ij}} \tag{11}$$

$$\implies \sum_i z_{ij} = \sum_i \frac{x_{ij}}{\sum_i x_{ij}} = 1$$

Thus, from (11), the objective function is as follows -

$$\text{Obj} = \sum_i \sum_j x_{ij}z_{ij} = \sum_i \sum_j x_{ij}(\frac{x_{ij}}{\sum_i x_{ij}}) = \sum_j \sum_i \frac{x_{ij}^2}{\sum_i x_{ij}} = \sum_j \frac{\sum_i x_{ij}^2}{\sum_i x_{ij}}$$

From the RMS (root mean square), arithmetic mean and standard deviation relation, we have -

$$x_{RMS}^2 = x_M^2 + \sigma_x^2 \tag{12}$$

$$\text{where} \quad x_{RMS} = \sqrt{\frac{x_1^2 + x_2^2 + \ldots + x_n^2}{n}} \tag{13}$$

$$x_M = \frac{x_1 + x_2 + \ldots + x_n}{n} \tag{14}$$

$$\sigma_x = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - x_M)^2} \tag{15}$$

On simplifying (12), (13), (14) and (15), we have -

$$\begin{aligned} \frac{\sum_i x_{ij}^2}{\sum_i x_{ij}} &= \frac{\sum_i x_{ij}}{n} + \frac{n\sigma_x^2}{\sum_i x_{ij}} \\ &= \frac{\sum_i x_{ij}}{n} + \frac{n(\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - x_M)^2)}{\sum_i x_{ij}} \end{aligned}$$

From constraint (2), we have $x_M = \frac{1}{m}$ and Theorem 1 states that there exists at least $n - d \cdot m$ elements having $x_{ij} = 1$ (i.e. on an average $\frac{n-d\cdot m}{m}$ per bin).
Thus,

$$\begin{aligned}
\frac{\sum_i x_{ij}^2}{\sum_i x_{ij}} &= \frac{\sum_i x_{ij}}{n} + \frac{n(\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \frac{1}{m})^2)}{\sum_i x_{ij}} \\
&= \frac{\sum_i x_{ij}}{n} + \frac{\sum_{i=1}^{n}(x_{ij} - \frac{1}{m})^2}{\sum_i x_{ij}} \\
&\geq \frac{\sum_i x_{ij}}{n} + \frac{\frac{n-dm}{m}(1 - \frac{1}{m})^2}{\sum_i x_{ij}} \\
&\geq \frac{\sum_i x_{ij}}{n} + \frac{\frac{n-dm}{m}(1 - \frac{1}{m})^2}{\frac{n}{m}}
\end{aligned}$$

$$\begin{aligned}
\implies \sum_j \frac{\sum_i x_{ij}^2}{\sum_i x_{ij}} &\geq \sum_j \frac{\sum_i x_{ij}}{n} + \sum_j \frac{\frac{n-dm}{m}(1 - \frac{1}{m})^2}{\frac{n}{m}} \\
&= \frac{\sum_{ij} x_{ij}}{n} + m\frac{\frac{n-dm}{m}(1 - \frac{1}{m})^2}{\frac{n}{m}} \\
&= \frac{n}{n} + m\frac{(n - dm)(1 - \frac{1}{m})^2}{n} \\
&\geq 1 + m\frac{(n - dm)(1 - \frac{2}{m})}{n} \\
&= 1 + m(1 - \frac{dm}{n})(1 - \frac{2}{m}) \qquad \text{(15a)} \\
&= m - 1 + \frac{dm}{n}(2 - m) \\
&\approx m - 1 \qquad (n >> dm)
\end{aligned}$$

Hence, the value of the objective function ($\approx m - 1$) is close to it's upper bound ($= m$) indicating the near optimal usage of the bins. Any basic feasible solution would provide a near optimal packing when $n >> d \cdot m$ ∎

Corollary 6. *For a fixed value of $\sum_i x_{ij}$, the minimum value of $\frac{\sum_i x_{ij}^2}{\sum_i x_{ij}}$ occurs when all x_{ij}'s are equal. Also, the minimum value is equal to x_{ij}.*

Proof. From equations (12), (13), (14) and (15), we have -

$$\begin{aligned}\frac{\sum_i x_{ij}^2}{\sum_i x_{ij}} &= \frac{\sum_i x_{ij}}{n} + \frac{n\sigma_x^2}{\sum_i x_{ij}} \\ &\geq x_M\end{aligned}$$

The equality holds only when all the x_{ij}'s are equal ■

Any basic feasible solution for the relaxed LP will not necessarily be integral. However, from Theorem 5, we have seen that any such solution will be close to the optimal integer solution. We now present an algorithm which will provide the near optimal integer solution.

3 Approximation algorithm and it's bounds

Algorithm 1 PackingVectors(P_n, d)

Require: An set of vectors $p_1, p_2, \ldots, p_n$; P_n.

```
(m, X) = SolveLP(P_n, d)
if $m \geq \frac{n}{2}$ then
    return $FirstFit(P_n, d)$
else if $m \leq \sqrt{\frac{n}{d}}$ then
    return $GreedyLP(P_n, X, d)$
else
    return $IterativePack(P_n, X, d)$
end if
```

Theorem 7. *Algorithm 1 provides a $\theta(1)$-optimal guarantee to the Vector Bin Packing problem (VBP).*

Proof. From the solution of the relaxed LP, the packing problem is classified into 3 cases-

Case 1 : $m \geq \frac{n}{2}$ The optimal solution for an integer program is a feasible solution for the corresponding relaxed linear program. Thus, the optimal solution for a linear program is lesser than the optimal solution for an integer program (in a minimization problem). Thus, we have that -

$$\begin{aligned} m &\leq \text{OPT} \\ m &\geq \frac{n}{2} \end{aligned}$$

Combining the above, we get -

$$\text{OPT} \geq m \geq \frac{n}{2}$$

Since the total number of bins cannot exceed the number of vectors n, any first fit algorithm in the worst case can have an approximation factor of 2.

Case 2 : $m \leq \sqrt{\frac{n}{d}}$
Since $m \leq \sqrt{\frac{n}{d}}$, we have that -

$$n \geq dm^2 = m \cdot dm$$

Let $n = k \cdot dm$ for some value of k. The equation (15a) in Theorem 5 is then -

$$\begin{aligned} Obj &\geq 1 + m(1 - \frac{dm}{n})(1 - \frac{2}{m}) \\ &= 1 + m(1 - \frac{1}{k})(1 - \frac{2}{m}) \\ &= m - 1 - \frac{(m-2)}{k} \end{aligned}$$

If $k = \Omega(m)$, the equation then becomes -

$$\begin{aligned} Obj &= m - 1 - O(1) \\ &\geq m - 2 \qquad (k \geq m) \end{aligned}$$

Thus, the value of objective function is $\geq m-2$, indicating a (near-)optimal utility of each bin. We use this knowledge and apply a greedy heuristic as given in Algorithm 2.

From Theorem 1, the algorithm completely packs a majority of the vectors into the bins (i.e. $n - dm \geq dm(m-1) \geq \frac{n}{2}$ for $m \geq 2$). Less than half of the vectors remain to be packed by the repeated iteration. Hence, the required bins $\leq$ 2-OPT.

Case 3 : $\frac{n}{2} \geq m \geq \sqrt{\frac{n}{d}}$
We find all the instances where $\sum_i x_{ij} z_{ij} \geq \frac{1}{2}$. From corollary 6, $\sum_i x_{ij} z_{ij} \geq \frac{1}{2}$ happens only when more than half of the non-zero values of x_{ij}'s are greater than $\frac{1}{2}$ or the standard deviation is fairly high. Also, the utility of the bin is more than $\frac{1}{2}$.

Only those bins whose utility exceeds half are chosen, and the values of x_{ij}'s are sorted in descending order. The values of x_{ij}'s less than half are ignored (less than half the number of non-zero x_{ij}'s). The value of standard

Algorithm 2 GreedyLP(P_n, X, d)

```
Require: An set of vectors $p_1, p_2, \ldots, p_n$; $P_n$ and a set of $x_{ij}$ values $X$.
$P_n^{'} = P_n$
$X^{'} = SortDescending(X)$
while $X^{'} \neq \Phi$ do
    Remove the top element $x_{ij}$ in $X^{'}$
    if vector $p_i$ fits in bin $j$ then
        $Pack(i, j)$
        $P_n^{'} = P_n^{'} \backslash (p_i)$
    end if
    $X^{'} = X^{'} \backslash \{x_{ij}\}$
end while
$PackingVectors(P_n^{'}, d)$
```

Algorithm 3 IterativePack(P_n, X, d)

```
Require: An set of vectors $p_1, p_2, \ldots, p_n$; $P_n$ and a set of $x_{ij}$ values $X$.
$P_n^{'} = P_n$
$Z = FindDualObj(X, d)$
for $j = 1$ to $m$ do
    if $\sum_i x_{ij} z_{ij} \geq \frac{1}{2}$ then
        $X_j^{'} = SortDescending(X_j)$
        $X_j^{''} = RemoveLessThanHalf(X_j^{'})$
        $Pack(X_j^{''})$
        $P_n^{'} = P_n^{'} \backslash PackedVectors$
    end if
end for
$PackingVectors(P_n^{'}, d)$
```

deviation can also be high in which case, we have achieved our objective of increasing the gap between the values of x_{ij}.

We now proceed to packing the vectors into their respective bins. Since the number of x_{ij}'s thrown away are less than half and the remaining values of x_{ij}'s are $\geq \frac{1}{2}$. Atmost twice the number of bins are used to pack these vectors with $x_{ij} \geq \frac{1}{2}$. The remaining vectors are deferred to the next iteration.

Thus, in each stage of the algorithm, the number of bins used is atmost twice the optimal since we are packing vectors with $x_{ij} \geq \frac{1}{2}$. Hence, we have proved that the number of bins used is atmost 2-OPT. ∎

References

[1] Nikhil Bansal and Alberto Caprara and Maxim Sviridenko. Improved approximation algorithms for multidimensional bin packing problems. *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society, Washington, DC, USA (2006) 697–708

[2] Nikhil Bansal and Jos R. Correa and Claire Kenyon and Maxim Sviridenko. Bin Packing in Multiple Dimensions: Inapproximability Results and Approximation Schemes. *Math. Oper. Res.* **31**(1), (2006) 31–49

[3] Alberto Caprara. Packing 2-Dimensional Bins in Harmony. *FOCS '02: Proceedings of the 43rd Symposium on Foundations of Computer Science*, IEEE Computer Society, Washington, DC, USA (2002) 490–499

[4] Alberto Caprara and Paola Toth. Lower bounds and algorithms for the 2-dimensional vector packing problem. *Discrete Appl. Math.* **111**(3), (2001) 231–262

[5] Chandra Chekuri and Sanjeev Khanna. On multi-dimensional packing problems. *SODA '99: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics, Baltimore, Maryland, USA (1999) 185–194

[6] George Bernard Dantzig and Mukund Narain Thapa. *Linear Programming: Theory and extensions*, Springer-Verlag (2003).

[7] Wenceslas Fernandez de la Vega and George S. Lueker. Bin packing can be solved within 1+epsilon in linear time. *Combinatorica* **1**(4), (1981) 349–355

[8] David Karger and Krzysztof Onak. Polynomial approximation schemes for smoothed and random instances of multidimensional packing problems. *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2007) 1207–1216

[9] Narendra Karmarkar and Richard M. Karp. An efficient approximation scheme for the one-dimensional bin-packing problem. *SFCS '82: Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, IEEE Computer Society, Washington, DC, USA (1982) 312–320

[10] Hans Kellerer and Vladimir Kotov. An approximation algorithm with absolute worst-case performance ratio 2 for two-dimensional vector packing. *Operations Research Letters* **31**(1), (2003) 35–41

[11] Boaz Patt-Shamir and Dror Rawitz. Vector Bin Packing with Multiple-Choice. *CoRR* **abs/0910.5599**, (2009) http://arxiv.org/abs/0910.559

[12] Gerhard J. Woeginger. There is no asymptotic PTAS for two-dimensional vector packing. *Information Processing Letters* **64**(6), (1997) 293–297

4 Appendix

4.1 Dual formulation of the ILP

$$\text{minimize}: \quad \sum_j y_j \tag{16}$$

$$\text{such that} \quad \sum_j x_{ij} = 1 \qquad 1 \le i \le n \tag{17}$$

$$\sum_i p_i^k.x_{ij} \le 1 \qquad 1 \le j \le m, 1 \le k \le d \tag{18}$$

$$y_j \ge x_{ij} \qquad 1 \le i \le n, 1 \le j \le m \tag{19}$$

$$x_{ij} \in \{0,1\} \qquad 1 \le i \le n, 1 \le j \le m \tag{20}$$

Multiply constraint (19) by positive multipliers z_{ij} corresponding to x_{ij}'s. Adding all such constraints, we obtain -

$$(\sum_i z_{ij})y_j \ge \sum_i x_{ij}z_{ij} \qquad 1 \le j \le m$$

$$\sum_j(\sum_i z_{ij})y_j \ge \sum_j \sum_i x_{ij}z_{ij}$$

Further, we have -

$$\sum_j y_j \geq \sum_j (\sum_i z_{ij}) y_j \geq \sum_j \sum_i x_{ij} z_{ij}$$
$$\text{subject to} \qquad \sum_i z_{ij} \leq 1 \tag{21}$$

Thus, the minimization problem can be reframed as a maximization problem with the constaint (21) and objective function being -

$$\max : \quad \sum_i \sum_j x_{ij} z_{ij}$$

Adding the new constraints and relaxing constraint (20), the dual problem is as follows -

$$\begin{aligned}
& \text{maximize :} \quad \sum_i \sum_j x_{ij} z_{ij} \\
\text{such that} \quad & \sum_j x_{ij} = 1 && 1 \leq i \leq n \\
& \sum_i p_i^k . x_{ij} \leq 1 && 1 \leq j \leq m, 1 \leq k \leq d \\
& \sum_i z_{ij} \leq 1 && 1 \leq j \leq m \\
& x_{ij}, z_{ij} \geq 0 && 1 \leq i \leq n, 1 \leq j \leq m
\end{aligned}$$